(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 942 569 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
13.10.1999 Bulletin 1999/41

(43) Date of publication A2:
15.09.1999 Bulletin 1999/37

(21) Application number: 99301555.1

(22) Date of filing: 02.03.1999

(51) Int Cl.^{6}: H04L 29/06

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 13.03.1998 US 39112

(71) Applicant: LUCENT TECHNOLOGIES INC.
Murray Hill, New Jersey 07974-0636 (US)

(72) Inventors:
- Doshi, Bharat Tarachand
  Holmdel, New Jersey 07733 (US)
- Dravida, Subrahmanyam
  Freehold, New Jersey 07728 (US)
- Hernandez-Valencia, Enrique
  Highlands, New Jersey 07732 (US)
- Matragi, Wassim A.
  Brooklyn, New York 11209 (US)
- Qureshi, Muhammed Akber
  Metuchen, New Jersey 08840 (US)

(74) Representative:
Watts, Christopher Malcolm Kelway, Dr. et al
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green Essex, IG8 0TU (GB)

(54) Simple data link (SDL) protocol

(57) A simple point-to-point data link protocol (SDL) is defined which is based on the use of a length indicator field and an error check field, rather than a flag, for performing packet boundary recovery in a receiver. In an embodiment of the invention, an SDL transmitter transmits SDL packets comprising a header and a variable length payload. The SDL header comprises a length indicator (LI) field, a type field and a cyclic redundancy check (CRC) field. For receiving these transmitted SDL packets, SDL supports the use of a self-synchronization/self-delineation technique in the receiver. The receiver performs self-delineation as a function of the LI field, and performs self-synchronization, or packet recovery, as a function of both the LI field and the header CRC field. In particular, in performing packet recovery, the receiver performs a CRC check over each received SDL packet header and synchronization is declared after *N* correct checks, e.g., *N* = *4*. The SDL receiver operates in a hunt mode when performing synchronization, and a normal mode when synchronization has been accomplished.

FIG. 1

LENGTH INDICATOR (L1) (16)	TYPE (6)	HEADER CRC (10)	PROTOCOL (8-16)	INFORMATION (≥0)	FCS (32)

Printed by Jouve, 75001 PARIS (FR)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 30 1555

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
P,X	CARLSON J ET AL: "PPP over Simple Data Link (SDL) using SONET/SDH with ATM-like framing" INTERNET DRAFT, ‘Online! November 1998 (1998-11), XP002112807 Retrieved from the Internet: <URL:ftp://ftp.cns.vt.edu/pub/internet/draft/draft-ietf-pppext-sdl-00.txt> ‘retrieved on 1999-07-23! * the whole document * -----	1-8	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	20 August 1999	Eraso Helguera, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 909 051 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.04.1999 Bulletin 1999/15

(51) Int Cl.6: H04J 3/14, H04L 25/03, H04J 3/06

(21) Application number: 98308085.4

(22) Date of filing: 05.10.1998

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 11.10.1997 US 947538

(71) Applicant: LUCENT TECHNOLOGIES INC.
Murray Hill, New Jersey 07974-0636 (US)

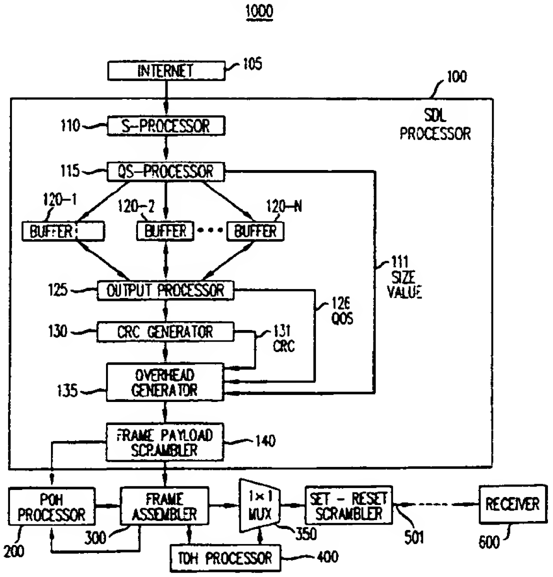
(72) Inventors:
- Anderson, Jon
Brielle, New Jersey 08730 (US)
- Dravida, Subrahmanyam
Freehold, New Jersey 07728 (US)
- Doshi, Bharat Tarachand
Holmdel, New Jersey 07733 (US)
- Hernandez-Valencia, Enrique
Highlands, New Jersey 07732 (US)
- Krishnaswamy, Murali
Piscataway, New Jersey 08854 (US)
- Manchester James S.
Freehold, New Jersey 07728 (US)

(74) Representative:
Watts, Christopher Malcolm Kelway, Dr. et al
Lucent Technologies (UK) Ltd,
5 Mornington Road
Woodford Green Essex IG8 OTU (GB)

(54) Simplified data link

(57) A simplified data link protocol which may be implemented in a very high-speed transmission system, e. g., SONET, processes a datagram received from an IP facility according to QoS considerations and scrambles a datagram before it is again scrambled by a transmission system, e.g., a SONET transmitter, to ensure that the pattern of a user's data does not match the transmission scrambling pattern. The data link protocol scrambler also employs a novel synchronization scheme. We also use a pointer system which identifies the location of a datagram in a frame to eliminate flags and the need to process user data to ensure that it does not contain and a boundary flag.

FIG.2



Printed by Jouve, 75001 PARIS (FR)

BRIEF DESCRIPTION OF THE DRAWING:

[0011]

FIG.1 illustrates the way in which prior data systems delineate the boundaries of a transmitted datagrams and packets;

FIG. 2 is a block diagram of a simplified data link transmitter system in which the principles of the invention may be practiced;

FIG. 3 is a layout of a SONET (STS-1) Synchronous Transport Signal Level 1;

FIG. 4 illustrates an alternative arrangement for building a Synchronous Payload Envelope bearing a plurality of STS frames;

FIG. 5 is a block diagram of a simplified data link receiver system in which the principles of the invention may be practiced;

FIG. 6 is a block diagram of the frame payload scrambler of FIG. 2;

FIG. 7 illustrates the format of a descrambling code that the frame payload scrambler of FIG. 2 inserts in the path overhead section of a SONET frame; and

FIG. 8 is a block diagram of the frame payload descrambler of FIG. 5.

DETAILED DESCRIPTION:

[0012] The Simplified Data Link (SDL) shown in FIG. 2 includes S-processor 110 which provides an interface for receiving a datagram from an Internet facility 115, such as an IP gateway (router), computer etc., and which determines the size (i.e., number of bytes) of the incoming datagram. The S-processor may do this by either (a) counting each byte forming the incoming datagram, or (b) checking the datagram header for such information if the datagram was formed in accordance with the so-called IP version IV protocol. For example, the IP version IV protocol includes the size of the datagram in the datagram header. If that is the case, then S-processor 110 may then simply query the datagram header. S-processor 110 then supplies via path 111 a value indicative of the size of the datagram to overhead generator 135, which appends that value and other Information to the accompanying datagram header, as will be explained below. The incoming datagram is then fed to QoS processor 115, which determines the level of priority that should be accorded to the incoming datagram. QoS processor 115 stores a datagram associated with the highest level of quality in data buffer 120-1; and stores a datagram associated with the next highest level of priority in data buffer 120-2 and so on. QoS processor 115 may determine such level of priority in a number of different ways, For example, if, as mentioned above, the datagram was formed in accordance with the IP version IV protocol, then the datagram header contains data indicative of the type of service associated with the datagram. If that is the case, then the datagram header may contain QoS properties. QoS processor 115 using either the identified type of service or QoS properties determines the level of priority associated with the datagram and stores the datagram in the appropriate one of the buffers 120-1 through 120-N. Note that one or more of the buffers 120-1 through 120-n may be a straight through path to output processor 125, as represented by the dashed line in buffer 120-1 --- meaning that the datagram is not stored in the buffer but is passed straight through the buffer to output processor 125.

[0013] Each of the buffers 120-1 through 120-N includes a scheduling processor (not shown) which schedules, on a priority type basis, for access to output processor 125. Thus, for example, if a number of the buffers contend for access to output processor 125 at the same time, then the buffer associated with the highest level of priority is granted such access. Specifically, each contention processor cancels its contention if it determines that a buffer of a higher priority is also contending for access to processor 125. Thus, output processor 125 receives the datagram from the buffer 120-i that wins such contention, and forwards the datagram as it is received to conventional CRC generator 130. Alternatively, processor 125 may receive a datagram from a buffer 120-i according to some other QoS scheduling policy.

[0014] Output processor 125 also forwards a value indicative of the QoS that is to be accorded to the datagram to overhead generator 135 via path 126. CRC generator 130, which may be, for example, a conventional high-speed processor/computer, generates a conventional CRC code across the contents forming the datagram and supplies the CRC to overhead generator 135 via path 131 and also supplies the datagram to overhead generator via path 132. Overhead generator 135, in turn, appends the information that it respectively receives via paths 111, 126 and 131 to the datagram header, all in accordance with an aspect of the invention. It then supplies the resulting datagram to frame payload scrambler 140.

[0015] As discussed above, the aforementioned synchronization process may be disrupted irrespective of the fact that a scrambler circuit used. As mentioned, a disruption may occur when the user's packet is larger than the scrambler period and when the pattern of the user's data matches the scrambling pattern. As was also discussed above, it is possible for a user to insert the scrambler pattern in the user's datagram and if those bits are aligned with the scrambler pattern, then the scrambler circuit would output a stream of zeroes (or all datagram will equal the generator 130 CRC value. If such CRC values are not equal, then processor 710 moves the boundaries (or window) covering what it hopes is the datagram by one bit and recalculates the CRC . If the latter CRC equals the generator 130 CRC, then processor 710 concludes that the new boundaries encompass the datagram. If not, then processor 710 again moves the boundaries by one bit and again recalculates the CRC. Processor 710 continues this process until the CRC that it calculates equals the CRC received in the POH. when that event occurs, then processor 710 knows such boundaries, and is thus able to verify the value of the length byte. Processor 710 then supplies the datagram to SDL overhead processor 720, which strips off the size and QoS bytes and supplies those values to paths 715 and 717, respectively. Processor 710 also supplies the datagram to QoS processor 720, which operates similar to QoS processor 115 (FIG. 2).

[0021] Specifically, (and similar to what has already been discussed in conjunction with FIG. 2) QoS processor 720 also determines the level of priority that should be accorded the datagram that it receives from processor 715, in which such priority is based on the QoS value that it receives via path 717. Similarly, QoS processor 720 stores a datagram associated with the highest level of quality in data buffer 725-1; stores a datagram associated with the next highest level of priority in data buffer 725-2 and so on. Similarly, one or more of the buffers 725-1 through 725-N may be a straight through path to output processor 730, as represented by the dashed line in buffer 725-1 --- meaning that the datagram is not stored in the buffer but is passed straight through the buffer to output processor 730.

[0022] Each of the buffers 725-1 through 725-N also includes a scheduling processor (not shown) which contend, on a priority type basis, for access to output processor 730. For example, if a number of the buffers contend for access to output processor 730, then the buffer associated with the highest level of priority is granted such access. Specifically, each contention processor cancels its contention if it determines that a buffer of a higher priority is also contending for access to processor 730. Thus, output processor 730 receives the datagram from the buffer 725-i that wins such contention, and forwards the datagram as it is received to a conventional interface buffer 635 that provides an interface between SDL receiver 700 and some other Internet facility, e.g., an Internet router. Alternatively, processor 730 may receive a datagram from buffer 725-i according to some other QoS scheduling policy.

[0023] A block diagram of the frame payload scrambler used in the SDL processor at the transmitter is shown in FIG. 6. Frame payload scrambler 800 includes scrambler section 810 comprising a shift register whose operation is characterized by the following polynomial:

$$1 + x^{2} + x^{19} + x^{21} + x^{40}$$

[0024] The polynomial function is implemented in scrambler 810 by a shift register formed from a plurality of registers 815-1 through 815-40 that are driven by a system clock signal (not shown) to generate, in conjunction with the adder circuits 820-1 through 820-3, a random and continuous pattern of logical ones zeros at the output 816 of register 815-1 (also shown as bit ao). The random, continuos stream of logical ones and zeroes is presented to one input of Exclusive Or (Ex Or) circuit 830 via an extension of path 816. The data (bits) that are to be scrambled are supplied to another input of Ex Or circuit 830 via path 825. The scrambled result of the Ex Or is then supplied to path 831. In Fig. 2, input path 825 extends from overhead generator 135 and output path 831 connects to one input of frame assembler 300. It is noted that scrambler 810 is initialized at start up using a 40 bit data word in which at least one bit must be a logical one (non-zero).

[0025] To synchronize the descrambler circuit 705 that is in the receiver 600 (FIG. 5) with the scrambler 810 that is in the transmitter, scrambler 800 predicts (also referred to herein as "projects") what the state of transmitter scrambler 810 (i.e., the scrambler code) will be a predetermined number of bytes in the future and supplies that prediction/determination to the receiver so that the SDL receiver may be properly synchronized with the transmitter and properly descramble a received scrambled payload. Such a determination is periodically transmitted to the receiver SDL. Accordingly, then, the receiver may quickly restore synchronization with the transmitter whenever such synchronization has been interrupted.

[0026] Since a SONET frame (specifically the path overhead) has a limited amount of unused data bytes that may be used to transmit the aforementioned prediction/determination, which comprises, for example, five bytes of data, the predicted descrambling code is transmitted over two consecutive frames as one embodiment. Thus, the receiver may be out of synchronization for, at most, two frames. (It is understood that the descrambling code could be transmitted over one frame if the appropriate number of byte locations were available. In that case, then, the receiver would be out of synchronization for one frame.) More specifically, the so-called H4, Z3 and Z4 bytes of the path overhead are used to transport the predicted state to the receiver, in which a CRC code generated over the five byte state is also sent in one of those path overhead bytes.

[0027] An illustrative format for the scrambling/descrambling code is shown in FIG. 7 and includes fields 70-1 through 70-5. Field 70-1 contains a start/begin bit set to a logical one followed by field 70-2 containing 23 bits of the scrambling code (state). Fields 70-1 and 70-2 comprise three bytes which are inserted in the aforementioned fields of the path overhead of the first transmitted frame. Field 70-3 contains an end bit and is followed by field 70-4 containing the remaining bits of the five byte code. A CRC generated over the five byte code agram being formed from a particular number of data bytes and being associated with a predetermined quality of service, said apparatus comprising

a simplified data link processor that receives the datagram from the Internet, assigns a size value to the datagram, the size value being determined as a function of the number of data bytes forming the datagram, and forwards the datagram for storage in a buffer, in which the buffer is selected as a function of a quality of service indicator indicative of a level of priority assigned to the datagram and in which the contents of the selected buffer is then processed at a level commensurate with the assigned priority level, the simplified data link processor including

first apparatus which generates a cyclic redundancy code (CRC) over the datagram and inserts the CRC, a quality of service indicator and size value in a header section of the datagram, and which then supplies the resulting datagram to second apparatus,

the second apparatus being operative for receiving the data bits forming the resulting datagram and scrambling the bits of the datagram in accordance with a particular code and then supplying the scrambled result to the Synchronous Optical NETwork apparatus, the Synchronous Optical NETwork apparatus including a frame assembler which assembles the scrambled result and path overhead information into a payload and a scrambler circuit which then scrambles the payload and associated transport overhead Information and outputs that scrambled result as a frame of data to an optical path.

2. The simplified data link apparatus of claim 1 wherein the second apparatus scrambling process is characterized by a polynomial of $1 + x^2 + x^{19} + x^{21} + x^{40}$.

3. The simplified data link apparatus of claim 1 wherein the Synchronous Optical NETwork apparatus further includes a path overhead processor which generates path overhead information that is included as part of the payload, the path overhead information including a descrambling code derived as a function of the particular code that was used by the second apparatus to scramble the datagram.

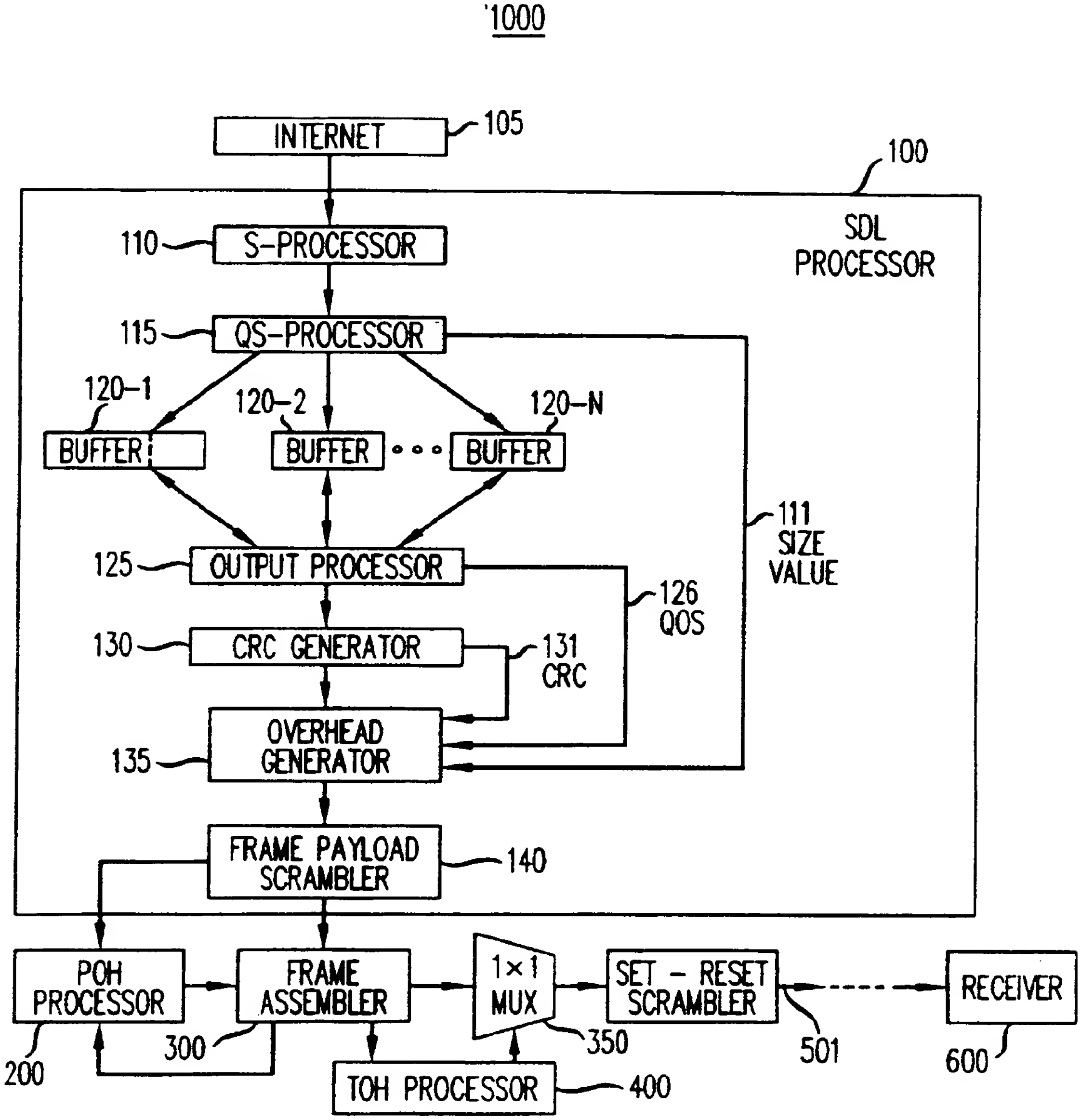
4. The simplified data link apparatus of claim 1 wherein the Synchronous Optical NETwork apparatus further includes a path overhead processor which generates path overhead information that is included as part of the payload, the path overhead information including a first part of a descrambling code derived as a function of the particular code that was used by the second apparatus to scramble the datagram and wherein the path overhead information contained in a next succeeding frame that is outputted to the optical path includes a second part of the second apparatus descrambling code.

5. Apparatus for processing a datagram received from Internet facilities and supplying the processed datagram to a facility which transmits data over a network at a very high data rate, said apparatus comprising

a first scrambler for scrambling the datagram bits starting with a current scrambling code to form a scrambled datagram and for determining a projected descrambling code that succeeds the current code by a predetermined number of bits and is operable for descrambling the datagram at a receiver, and for including the projected descrambling code in the transmission of the scrambled datagram to the receiver, and

at the receiver, receiving the scrambled datagram and forming the projected descrambling code into a scrambling code and descrambling the scrambled datagram.

FIG.2


1000
INTERNET
105
100
SDL PROCESSOR
110
S-PROCESSOR
115
QS-PROCESSOR
120-1
120-2
120-N
BUFFER
BUFFER
BUFFER
111 SIZE VALUE
125
OUTPUT PROCESSOR
126 QOS
130
CRC GENERATOR
131 CRC
135
OVERHEAD GENERATOR
FRAME PAYLOAD SCRAMBLER
140
POH PROCESSOR
FRAME ASSEMBLER
1×1 MUX
SET - RESET SCRAMBLER
RECEIVER
200
300
350
501
600
TOH PROCESSOR
400

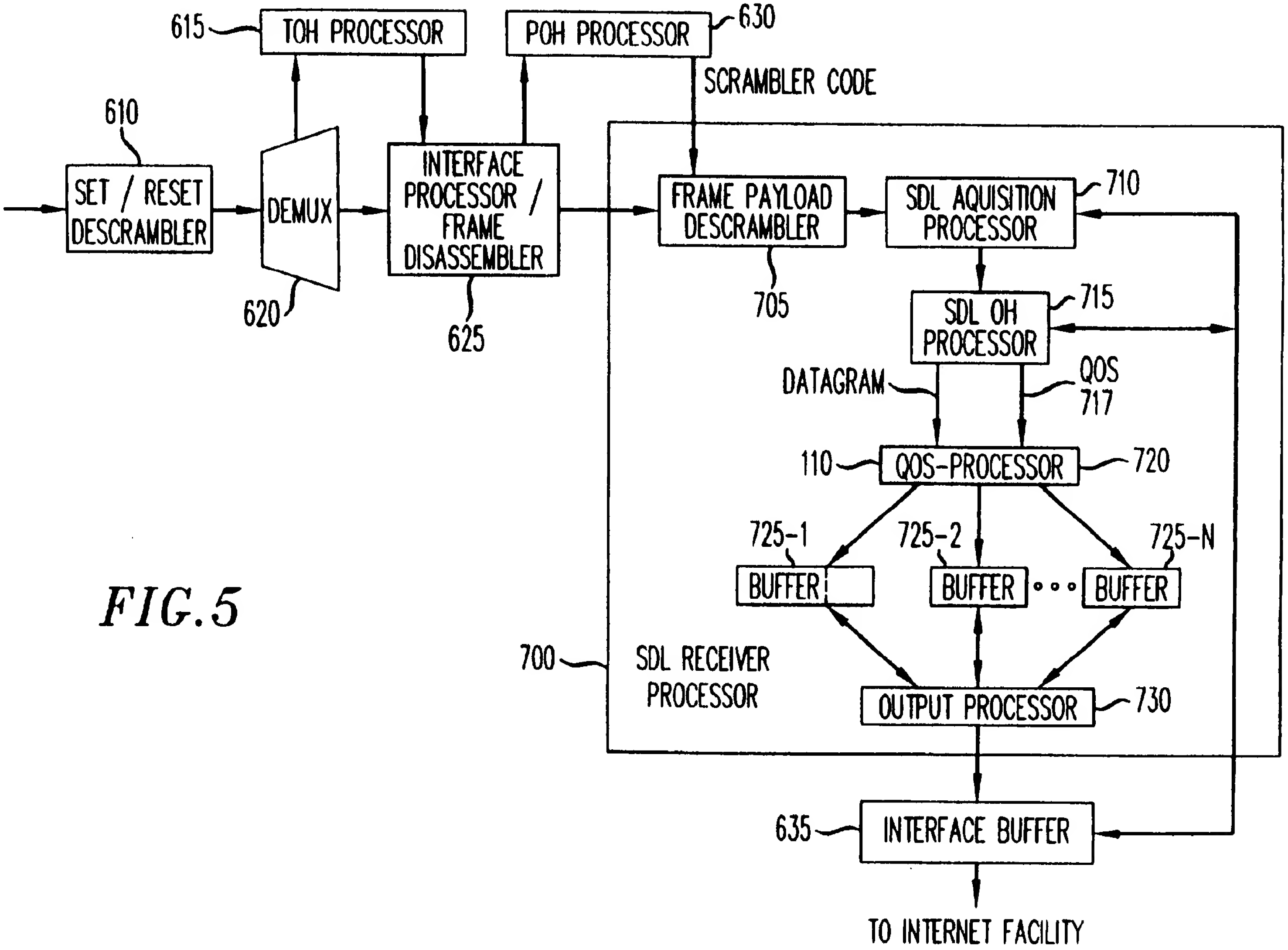
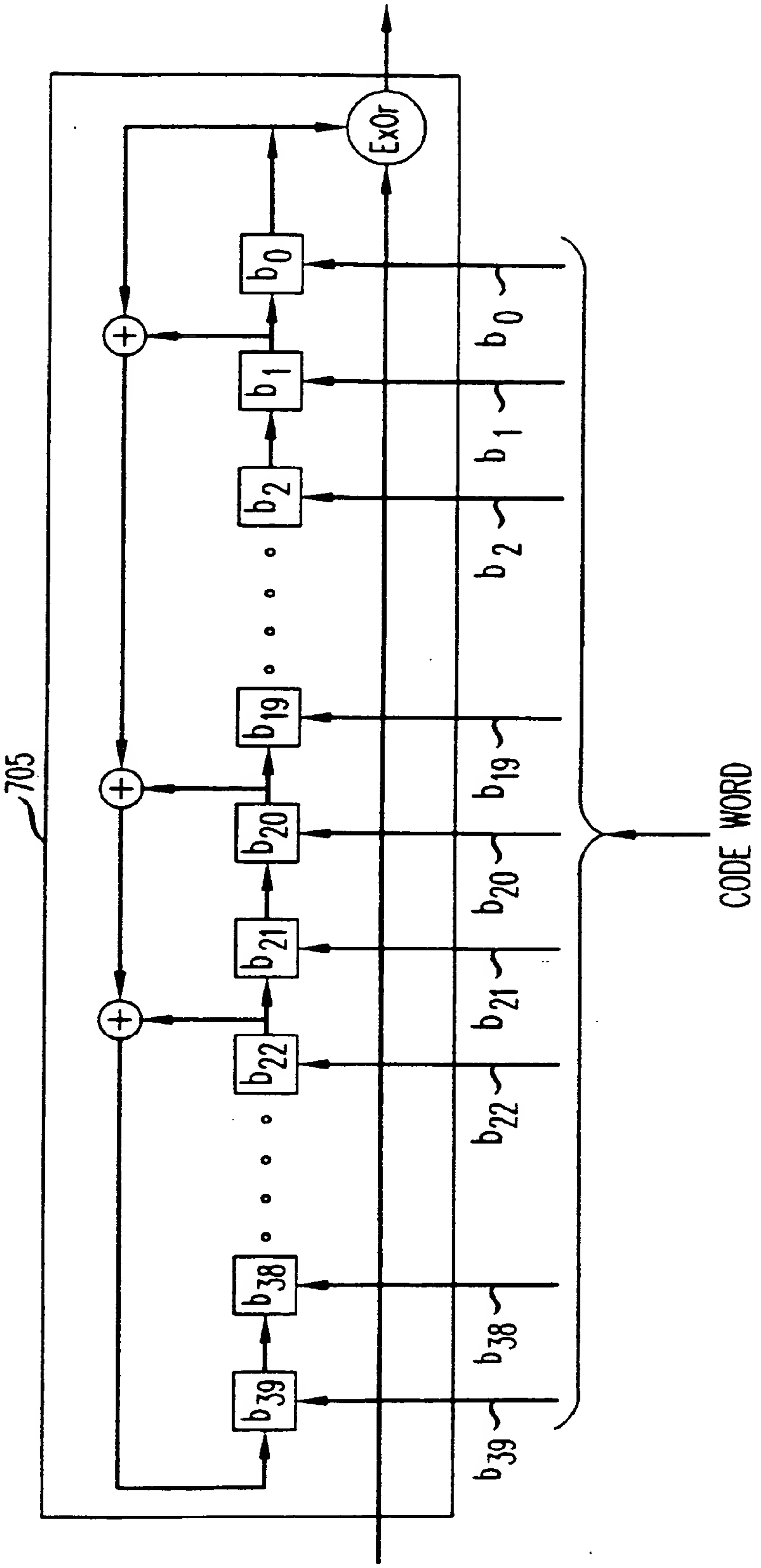

610
SET / RESET DESCRAMBLER
615
TOH PROCESSOR
DEMUX
620
INTERFACE PROCESSOR / FRAME DISASSEMBLER
625
POH PROCESSOR
630
SCRAMBLER CODE
FRAME PAYLOAD DESCRAMBLER
705
SDL AQUISITION PROCESSOR
710
SDL OH PROCESSOR
715
DATAGRAM
QOS
717
110
QOS-PROCESSOR
720
725-1
BUFFER
725-2
BUFFER
725-N
BUFFER
700
SDL RECEIVER PROCESSOR
OUTPUT PROCESSOR
730
635
INTERFACE BUFFER
TO INTERNET FACILITY

FIG.5

FIG. 8


705
ExOr
+
+
+
b_0
b_1
b_2
b_{19}
b_{20}
b_{21}
b_{22}
b_{38}
b_{39}
b_0
b_1
b_2
b_{19}
b_{20}
b_{21}
b_{22}
b_{38}
b_{39}
CODE WORD